Appl. No. 09/397,455 Attorney Docket: 042390.P6764

In The United States Patent and Trademark Office
Before The Board of Patent Appeals and Interferences

In re Patent Application of:	Smith, Ned M.	Examiner:	Fields, Courtney D.
		Art Unit:	2132
Application No.:	09/397,455		
Filed:	September 16, 1999		
For:	METHOD AND APPARATUS TO ASSIGN TRUST TO A KEY		

APPEAL BRIEF
IN SUPPORT OF APPELLANTS' APPEAL
TO THE BOARD OF PATENT APPEALS AND INTERFERENCES

Honorable Director of the United States Patent and Trademark Office
Washington, DC 20231

Sir/Madam:

Applicant (hereafter "Appellant") hereby submits this Brief in support of their Appeal from a final decision by the Examiner in the above-captioned case. Appellant respectfully requests consideration of this Appeal by the Board of Patent Appeals and Interferences for allowance of the claims in the above-captioned patent application.

An oral hearing is not desired.

TABLE OF CONTENTS

1. REAL PARTY IN INTEREST

The invention is assigned to Intel Corporation of 2200 Mission College Boulevard, Santa Clara, California 95052.

2. RELATED APPEALS AND INTERFERENCES

To the best of Appellants' knowledge, there are no appeals or interferences related to the present appeal that will directly affect, be directly affected by, or have a bearing on the Board's decision.

3. STATUS OF THE CLAIMS

Claims 1, 2, and 4-26 are now pending in the above referenced patent application. Claims 1, 2, and 4-26 were rejected in the Final Office Action mailed on June 24, 2005 and are the subject of this appeal.

4. STATUS OF THE AMENDMENTS

No amendments have been filed subject to the Final Rejection.

A copy of all claims on appeal is attached hereto as Appendix A.

5. SUMMARY OF THE CLAIMED SUBJECT MATTER

In modern computing environments, it is desirable to identify the authenticity, integrity and authority of software modules seeking access to data/or and services for which access may be restricted. One technique for providing security is to associate a secret value, sometimes called a key, with each software module seeking access. If the possessor of the key may be traced back to a trusted source, such as, for example a "Certificate Authority" such as Verisign Inc. of Mountain View, California, the module or modules associated with the key may be trusted with access to select services and data.

One difficulty with this approach is that keys may be "compromised", meaning that secret components of their value may become known to a third party not intended to possess such knowledge. In well-known public-private key systems, such as the RSA Public Key Cryptosystem (1977), secret values may be compromised in a number of ways, including through inadvertent disclosure of the private key, or through reverse engineering (sometimes known as key or code "cracking") of data or software encrypted with the key.

When a key is compromised, the parties with unauthorized access to the key may impersonate authorized parties to obtain access to the secure services or data available to those with legitimate knowledge of the key. Consequently, it may be desirable to "revoke" the trusted status of the key so that it may no longer be used for access to secure data and services. Once revocation occurs, it may be difficult or impossible for software modules authorized to rely on the key to continue accessing the secure data or services because, along with the unauthorized parties, their access is revoked along with the trusted status of the key.

Software modules relying upon the revoked keys may embed an identification of the revoked key within the binary file or files comprising the modules themselves. In this

circumstance, the software module may be re-compiled, re-linked, and redistributed with a new embedded key whose trust has not been compromised. Recompilation, re-linkage, and redistribution of software modules may be an arduous and expensive process. Therefore, there exists a continuing need for techniques to assign trust in a new key once trust in a key has been compromised. (Appellant's Specification, pages 2 & 3.)

In this description, the private component of a public/private key pair may be referred to as the "private key" and the public component may be referred to as the "public key". Typically, in a manner well-known in the art of digital security, a "digital signature" may be generated by computing the hash value of a body of information, such as a data document or a sequence of program instructions, and then applying the private key to transform the generated hash value. A signature may be "verified", that is, determined to have been generated by the party or parties associated with the private key, by computing a second hash value on the body of information, then applying the public key component to the encrypted hash value corresponding to the private key value and comparing the hash values for a match. If the hatch values match, the recipient of the signature may have confidence that the signed body of information originated from a party associated with the private key and, further, that the body of information is unaltered from its state when the signature was generated. Again, methods of generating and verifying signatures are well-known in the art of digital security.

A "manifest" typically comprises one or more files containing attributes of another data file or software module. The manifest may typically comprise a hash value of the other file and an identification of the key used to sign the manifest. Using the identified key, a signature may be generated on the manifest. When the key used to sign the manifest is trusted, the integrity of the information comprised by the signed manifest may also be trusted. In order to provide a

measure of trust in the signing key, it may be possible to trace the signing key through a "certificate chain" back to a trusted source. This trusted source may be a key issued by a trusted party, such as a Certificate Authority (CA). Verisign Inc. is an example of a CA. The certificate chain may comprise one or more "digital certificates", that is, files comprising a key which are signed by keys which are closer in the chain of trust to the trusted source. For example, a bank may be assigned a private key for performing digital signatures. This private key may be assigned to the bank by a CA. A certificate for the bank's key may be issued by the CA, in the form of a file comprising the bank's key, signed by the CA's key. The CA's key may be described as the "anchor" of the trust extending to the bank's key. Certificates and certificate chains are well known by those of ordinary skill in the art of digital security; see, for example, Recommendation X.509 V.3 (1994) from the International Telecommunication Union.

The certificate chain may be comprised by the manifest for a software module, or it may be stored separately from the manifest. The hash value comprised by the manifest may be used to verify the integrity of the software module with which the manifest is associated, and may also serve to associate the manifest with the software module by utilizing the unique character of the hash value.

In one embodiment, one software module may control access to secure data and services in a computer system. This module may be called the security manager (SM). When another module in the computer system requests access to secure data or services, the SM may verify the integrity and trusted status of (henceforth referred to as cross-checking) the other software module (henceforth referred to as the client module). For example, when a client module requests access, the SM may consult the manifest for the client module. A hash value for the client module may be stored in this manifest. The SM may generate a hash value of the client

module and compare it with the hash value stored in the manifest. A match provides an indication regarding the integrity of the module to ensure that the module has not been tampered with. The SM may perform a hash on the manifest itself and compare it with the hash value comprised by the manifest signature. If those values match, they provide an indication of the integrity of the manifest itself and, hence, the hash value comprised by the manifest, providing further verification of the client module's integrity.

To determine whether the client module is associated with a trusted source, the SM may read from the client module manifest an identification of the public key component corresponding to the private key used to sign the manifest. Using this public key component, the SM may trace the association of the signing key back to the trusted source using a certificate chain. The certificates of the certificate chain may be comprised by the manifest or may be accessible separately from the manifest. For example, in one embodiment the key for the trusted source, to which the certificate chain traces back to, may be embedded within the binary file comprising the SM.

The SM may have an associated manifest similar to the manifest for the client module. In one embodiment, the manifest for the SM may comprise a hash value for the SM which the SM may use to verify its own integrity (self-check) in a manner similar in the manner in which the SM determines the integrity of the client module (cross check). The SM manifest may further comprise the public key component corresponding to the private key used to sign the SM manifest which, in a manner similar to the pubic key for the client module, may be associated with a trusted source through a certificate chain. The SM manifest may or may not comprise the certificate chain. The public key associated with the trusted source may be the same public key

to which the public key of the client module was traced through the client certificate chain. As previously noted, this public key component may be embedded in an SM binary file.

Figure 1 shows one embodiment 100 of a process to produce an SM module and the manifest for the SM module. The instructions 106 comprising the SM module, and a data file 110 may be input to a compilation/linking tool 120 (the compiler and linker may comprise separate tools). The data file may comprise a primary key 102 and one or more backup keys 104. In one embodiment, the primary key 102 and backup keys 104 comprise public keys which are trusted by the SM for performing secure operations, such as accessing secure data and services on a computer system. The compiler/linker 120 may output an SM binary image 130 suitable for loading into the memory of a computer system, and comprising the instructions 106 of the SM in binary form, e.g. machine language, and further comprising the key values from the data file 110 embedded within the binary image 130. In one embodiment, the key values 102,104 are embedded in a data area of the SM binary 130. In another embodiment, the key values 102,104 may be encoded in such a manner as to make their detection more difficult by unauthorized third parties examining the binary image 130, such as may take place with the aid of debugging or disassembly tool.

The SM binary 130 may be input to a hash generator 140 to generate a hash value unique to the SM binary 130 with a high degree of confidence. In other words, it would be highly improbable that another input to the hash generator 140 would result in the same hash value. In one embodiment, this hash value may be archived by a trusted authority, such as a Certificate Authority, to be employed in a manner to be described later. The hash value, along with a private key 160, may be input to a manifest generator 150. Other information (not shown) to be comprised by the manifest may also be input to the manifest generator 150. The manifest

generator 150 may output a manifest for the SM signed with the private key 160. In one embodiment, in addition to comprising the hash value of the SM, the manifest may comprise an identification of the public key corresponding to the private key 160 or a hash of this public key. The private key 160 may be traceable, by way of a certificate chain (the dotted line in Figure 1), back to the primary key 102 embedded in the SM binary 130. The certificate chain may be comprised by the manifest or separate from it.

Figure 2 shows an embodiment 200 of a process to produce the manifest for a client module of the SM, in other words, a software module to request secure data or services from the SM. (Appellant's Specification, pages 5-8.)

Embodiments in accordance with the present invention may employ manifest technology to provide a new trusted key to replace a compromised trusted key, such as a compromised primary key 102. Using such embodiments, it may be possible to revoke a compromised key without recompiling and redistributing the software modules that rely upon the compromised key. It may not be necessary to provide third parties, such as Certificate Authorities, with access to the source or binary code for these software modules when a key is compromised. Instead, these third parties may archive a hash of the software modules, which may then be distributed along with a new trusted key in a signed manifest. Because the archived hash is comprised by a manifest signed by a new trusted key, software modules can trust the integrity of the archived hash value when performing self or cross checks for module integrity.

When a key is compromised, a revocation manifest may be issued, for example by a Certificate Authority responsible for maintaining and managing trust in the compromised key value. In one embodiment, the revocation may be issued in the form of a new manifest

comprising an identification of a replacement key to be relied upon by software modules previously relying upon the compromised key. (Appellant's Specification, pages 9 & 10.)

Figure 3 shows an embodiment 300 of a process to produce a revocation manifest identifying compromised keys. Figure 4 is a flow chart illustrating an embodiment of a process by which trust may be assigned to a key relied upon by a software module to self-check itself or cross-check other modules, in accordance with the present invention. Figure 5 shows an embodiment of an apparatus to assign trust in a new key. (Appellant's Specification, pages 10 & 11.)

6. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The above referenced patent application has been reviewed in light of the Office Action, dated June 24, 2005, in which:

- claims 1, 2, and 4-26 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Grimmer (US Patent No. 5,774,552) in combination with Van Oorschot (US Patent No. 6,215,872 B1).

7. ARGUMENT

7.1. 35 U.S.C. § 103(a)

7.1.1. Grimmer and Van Oorschot: Claims 1, 2, and 4-26

The PTO has also rejected claims 1, 2, and 4-26 under 35 U.S.C. § 103(a) based upon Grimmer in combination with Van Oorschot. The rejection of these claims is respectfully traversed.

M.P.E.P. § 706.02(j) sets forth the standard for a § 103(a) rejection:

> To establish a prima facie case of obviousness, three basic criteria must be met.
>
> First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or combine reference teachings.
>
> Second, there must be a reasonable expectation of success.
>
> Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations.
>
> The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on applicant's disclosure. *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991) (whitespace added).

Appellants respectfully assert that the combination set forth by the PTO fails to meet the requirement for a *prima facie* case for a § 103(a) rejection for at least the following reasons.

Appellant begins with claim 1. Claim 1 recites:

> 1. (Previously Presented) A method comprising:
>
> reading from a software module embedding one of a set of key associated with a trusted source;
>
> determining whether a key is traceable to one of the set of keys;
>
> determining whether the key is identified in a list of compromised keys; and
>
> if the key is not identified as compromised and is traceable to one of the keys in the set, assigning the key a trusted status.

7.1.1.1 Deficiency of the Cited Art

Appellants respectfully assert that the combination set forth by the PTO fails to meet the requirement for a *prima facie* case for a § 103(a) rejection for at least the following reasons.

It is respectfully asserted that neither Grimmer nor Van Oorschot, either alone or in combination, suggests or describes all the elements and limitations of claim 1.

It is respectfully asserted that Grimmer does not show, teach, use, or describe a reading **from a software module embedding one of a set of key** associated with a trusted source. Grimmer instead shows reading a set of keys **from a database** (the Certificate Authority, or LDAP of Fig. 6). It is respectfully asserted that Grimmer shows using a software module **to perform encryption**.

Appellant respectfully asserts that the public keys of Grimmer and Van Oorchot are **not embedded within a software module**. Grimmer instead shows **the public key being stored within an external database**. See Column 6, lines 36-38, and Fig. 6.

Appellant respectfully asserts that Van Oorchot does not ameliorate this deficiently. Van Oorchot instead shows public keys **stored within a database**, which is **accessed** by the security manager (software module). Appellant respectfully asserts that Figure 1 of Van Oorchot clearly shows that the security managers (Fig. 1, elements 12-16) are **separate and distinct** from the database directory (Fig. 1, element 18). It is also noted that Van Oorchot describes the security managers as separate "personal computers" that share the common directory. (See, Col. 3, lines 60-67.) Therefore, the common directory can not be embedded within each of the security managers.

Therefore, even if the combination were proper, although Appellant believes that it is not, nonetheless, the combination would still fail to produce the invention as recited in the rejected claims. It is, therefore, respectfully requested that the rejection of this claim be withdrawn.

7.1.1.2 Response to PTO's Remarks: Limitations of the Claims

In the June 24, 2005 office action the PTO stated that "features upon which applicant relies ... are not recited in the rejected claims." See the June 2005 Office Action, page 2, lines 16-19. Appellant respectfully disagrees.

For convenience, Appellant repeats claim 1. Claim 1 recites:

> 1. (Previously Presented) A method comprising:
>
> reading from a software module embedding one of a set of key associated with a trusted source;
>
> determining whether a key is traceable to one of the set of keys;
>
> determining whether the key is identified in a list of compromised keys; and
>
> if the key is not identified as compromised and is traceable to one of the keys in the set, assigning the key a trusted status.

The PTO states "The limitation does not clearly point out ... that the key is **read from the software module**." Appellant respectfully notes that claim 1, line 2 clearly states "**reading from a software module**." Appellant respectfully notes that, in the above quoted statement from the Office Action, the PTO uses the exact same words (allowing for "read" versus "reading") found in the Appellant's claim to argue that the limitation is not in Appellant's claim. Appellant asserts that the clear plain meaning of the cited phrase leaves little doubt that the limitation is "reading from a software module" and that no interpretation of the specification is needed. Appellant respectfully asserts that, as described above, this unmet limitation alone is sufficient to illustrate that the combination fails to produce the invention as recited in the rejected claims.

7.1.1.3 Response to PTO's Remarks: Only One Operation

Also, in the June 24, 2005 office action the PTO stated that "The limitation does not clearly point out that a key is embedded within a software module ...". The PTO further suggests that the claim suggests two distinct operations (1) reading from a software module, and (2) embedding a key. See the June 2005 Office Action, page 2, lines 17-20. Appellant respectfully disagrees.

7.1.1.3.1 Claim interpretation standard

MPEP § 2111 sets forth the standard for claim interpretation during examination.

> During patent examination, the pending claims must be "given the broadest reasonable interpretation consistent with the specification. ... The broadest reasonable interpretation of the claims must also be consistent with the interpretation that those skilled in the art would reach. *In re Cortright*, 165 F.3d 1353, 1359, 49 USPQ2d 1464, 1468 (Fed. Cir. 1999).

MPEP § 2111.01 further sets forth the standard of claim interpretation in both the case where a term is not defined in the specification and the case where the term is defined in the specification.

> When not defined by applicant in the specification, the words of a claim must be given their plain meaning. In other words, they must be read as they would be interpreted by those of ordinary skill in the art. *In re Sneed*, 710 F.2d 1544, 218 USPQ 385 (Fed. Cir. 1983)
>
> MPEP § 2111.01, 8th edition, 3rd paragraph.

> During examination, the claims must be interpreted as broadly as their terms reasonably allow. This means that the words of the claim must be given their plain meaning unless applicant has provided a clear definition in the specification. *In re Zletz*, 893 F.2d 319, 321, 13 USPQ2d 1320, 1322 (Fed. Cir. 1989).
>
> MPEP § 2111.01, 8th edition, 1st paragraph.

MPEP § 2111 further sets forth that the Appellants may act has their own lexicographer, so long as the definition is not repugnant to the "plain meaning" of the defined term.

> Applicant may be his or her own lexicographer as long as the meaning assigned to the term is not repugnant to the term's well known usage. *In re Hill*, 161 F.2d 367, 73 USPQ 482 (CCPA 1947).

7.1.1.3.2 Claim interpretation as applied to Claim 1

For convenience, Appellant repeats claim 1. Claim 1 recites:

1. (Previously Presented) A method comprising:

reading from a software module embedding one of a set of key associated with a trusted source;

determining whether a key is traceable to one of the set of keys;

determining whether the key is identified in a list of compromised keys; and

if the key is not identified as compromised and is traceable to one of the keys in the set, assigning the key a trusted status.

Appellant asserts that using a plain meaning approach the element of lines 2 & 3 is clearly one step. If it was two separate steps there would be a semicolon separating the steps as seen by the transition of the elements of line 3 to line 4, line 4 to line 5, and line 5 to line 6. A semicolon is used to connect **independent** clauses; therefore, if "reading" and "embedding" are not separated by a semicolon they must be **dependent** upon each other. Furthermore, each of these new, separate, and distinct elements (see the line citations above) are separated by not only a semicolon, but also a line break and an indentation. None of these three transitional devices separate the word "module" from the word "embedding" in line 2; therefore, it is asserted that a proper reading of the element leads one to believe that it is one element not two as suggested by the PTO.

Furthermore, if the key is read "**from** a software module" (as opposed to "**by** a software module") logic dictates that the key must be contained within the software module. If the key is not within the software module, it is not possible to read the key **from** the module. Therefore, reading the claim using nothing more than the logical plain meaning of the terms (as required by M.P.E.P. § 2111) leads one to understand that the key, which is read **from** the module, is embedded within the module. Once again, no interpretation of the specification is needed.

Appellant respectfully asserts that, as described above, this unmet limitation alone is sufficient to illustrate that the combination fails to produce the invention as recited in the rejected claims.

7.1.1.4 Conclusion

Therefore, even if the combination were proper, although Appellant believes that it is not, nonetheless, the combination would still fail to produce the invention as recited in the rejected claims. It is, therefore, respectfully requested that the rejection of this claim be withdrawn.

Claims 2 and 4-26 either depend from and include the limitations of claim 1, or include a substantially similar and patentably distinct limitation as claim 1. Therefore, these claims patentably distinguish from the cited patents on the same basis as claim 1. It is, therefore, respectfully requested that the PTO withdraw the rejections of these claims.

8. CONCLUSION

In view of the foregoing, it is respectfully asserted that all claims pending in this application, as amended, are in condition for allowance. If the Examiner has any questions, they are invited to contact the undersigned at 503-264-7002. Reconsideration of this patent application and early allowance of all claims is respectfully requested.

Respectfully submitted,

/s/Justin B. Scout/Reg. No. 54,431/
Justin B. Scout
Reg. No. 54,431

Dated: June 8, 2006

c/o Blakely, Sokoloff, Taylor & Zafman, LLP
12400 Wilshire Blvd., Seventh Floor
Los Angeles, CA 90025-1026
(503) 264-0967

APPENDIX A: CLAIMS APPENDIX

1. (Previously Presented) A method comprising:

reading from a software module embedding one of a set of key associated with a trusted source;

determining whether a key is traceable to one of the set of keys;

determining whether the key is identified in a list of compromised keys; and

if the key is not identified as compromised and is traceable to one of the keys in the set, assigning the key a trusted status.

2. (Original) The method of claim 1 further comprising:

verifying the integrity of a document comprising the key and the list of compromised keys.

3. (Cancelled)

4. (Original) The method of claim 1 in which determining whether the key is traceable to one of the set of keys further comprises:

tracing the key through a certificate chain to one of the keys in the set of keys.

5. (Original) The method of claim 1 further comprising:

associating a document comprising the key and the set of keys with a software module comprising the set of keys using a hash of the software module in the document.

6. (Original) The method of claim 2 in which the document is a manifest signed by the key.

7. (Original) The method of claim 1 in which determining whether the key is identified in the list of compromised keys further comprises:

searching the list of compromised keys for the key.

8. (Original) A method comprising:

producing a document comprising an identification of a software module and a list of compromised keys; and

digitally signing the document using a key traceable to one of a set of keys comprised by the software module.

9. (Original) The method of claim 8 in which the identification of the software module comprises a hash value of the software module.

10. (Original) The method of claim 8 in which the key is traceable to one of the set of keys comprised by the software module by way of a certificate chain.

11. (Original) The method of claim 8 further comprising:

making the document available on a communication network by which computer systems comprising the software module may read the document.

12. (Original) The method of claim 8 in which the set of keys is embedded within the software module.

13. (Original) A device comprising:

a processor;

a machine-readable storage medium coupled to the processor by way of a bus, the storage medium storing instructions which, when executed by the processor, cause the device to determine whether a key is traceable to one of a set of keys associated with a trusted source;

determine whether the key is identified in a list of compromised keys; and

if the key is not identified as compromised and is traceable to one of the keys in the set, assign the key a trusted status.

14. (Original) The device of claim 13 in which the instructions, when executed by the device, further cause the device to:

verify the integrity of a document comprising the key and the list of keys.

15. (Original) The device of claim 13 further comprising a software module comprising the list of keys.

16. (Original) The device of claim 13 in which the instructions, when executed by the device, further cause the device to:

trace the new key through a certificate chain to one of the keys in the set of keys.

17. (Original) A device comprising:

a processor;

a machine-readable storage medium coupled to the processor by way of a bus, the storage medium storing instructions which, when executed by the processor, cause the device to:

produce a document comprising an identification of a software module and a list of compromised keys; and

digitally sign the document using a key traceable to one of a set of keys comprised by the software module.

18. (Original) The device of claim 17 in which the identification of the software module comprises a hash value of the software module.

19. (Original) The device of claim 17 in which the key is traceable to one of the set of keys comprised by the software module by way of a certificate chain.

20. (Previously Presented) An article comprising a machine-readable medium having stored thereon instructions which, when executed by a processor, result in:

reading from a software module embedding one of a set of key associated with a trusted source;

determining whether a key is traceable to one of the set of keys;

determining whether the key is identified in a list of compromised keys; and

if the key is not identified as compromised and is traceable to one of the trusted keys, assigning the key a trusted status.

21. (Original) The article of claim 20 in which the instructions, when executed by the processor, further result in:

verifying the integrity of a document comprising the key and the list of keys.

22. (Original) The article of claim 20 further comprising a software module embedding the set of keys associated with the trusted source.

23. (Previously Presented) The article of claim 20 in which the sequence of instructions, when executed by the processor, further result in:

tracing the key through a certificate chain to one of the keys in the set of keys.

24. (Original) An article comprising a machine-readable medium having stored thereon instructions which, when executed by a processor, result in:

producing a document comprising an identification of a software module and a list of compromised keys; and

digitally signing the document using a key traceable to one of a set of keys comprised by the software module.

25. (Original) The article of claim 24 in which the identification of the software module comprises a hash value of the software module.

26. (Original) The article of claim 24 in which the key is traceable by way of a certificate chain to one of the set of keys embedded in the software module.

APPENDIX B: EVIDENCE APPENDIX

To the best of Appellants' knowledge, there is no evidence submitted pursuant to 37 C.F.R. §§ 1.130, 1.131, or 1.132 or of any other evidence entered by the examiner and relied upon by appellant in the present appeal that will directly affect, be directly affected by, or have a bearing on the Board's decision.

APPENDIX C: RELATED PROCEEDINGS APPENDIX

To the best of Appellants' knowledge, there are no appeals or interferences related to the present appeal that will directly affect, be directly affected by, or have a bearing on the Board's decision.